CLAIMS

1. Process for generating a performance model from a functional model for a system comprising a plurality of distributed hardware and software entities that engage to provide a service to at least one user, characterised in that it comprises the following stages:

- Distributing the representative system requests in a finite number of groups and identifying, for each request group, the corresponding execution flow, the distribution of said requests being determined by the service being called upon and by the characteristics of the customer specific behaviour, and the execution flow for each request group corresponds to the software entity execution linking, in sequence and/or in parallel, induced by a group request,

- Formalising the execution flows using a notation making it possible to highlight, on the one hand, the causal relationships between the different software entities of the system that are involved in the execution flows, and on the other hand, the information characterising the system's resource consumption, such as the period of time the CPU is occupied when a software entity is active,

- Developing an intermediate model that additionally comprises formalised execution flows, a resource specification that specifies the physical hardware of the system and an environment specification that represents user behaviour,

- Automating the conversion of the developed intermediate model into a performance model.

2. Process according to claim 1, characterised in that the performance model derived from the developed intermediate model is dedicated to pre-existing software simulators using queuing network techniques.

3. Process according to claim 1, characterised in that the distribution of system requests in a finite number of request groups is determined by the service being called upon, and by the characteristics of the customer specific behaviour which affect the way in which the service being called upon is delivered.

4. Process according to one of claims 1 to 3, characterised in that the execution flow for each request group is determined by the software entity execution linking, in sequence and/or in parallel, induced by a group request.

5. Process according to claim 4, characterised in that the topology of the queuing model derived from the conversion is wholly determined by the execution flows corresponding to the request groups.

6. Process according to claim 4, characterised in that the derivation of a performance model dedicated to a pre-existing simulator based on queuing network techniques can be automated by adapting the correspondence rules proposed.

7. Process according to one of claims 1 to 6, characterised in that the formalism of the phases is achieved using an extension of the MSC (Message Sequence Charts) formalism.

8. Process according to one of claims 1 to 7, characterised in that the formalisation of the graph of the phases and execution flows of a service using the HMSC (High level Message Sequence Charts) formalism is represented in the form of a tree comprising:

- a plurality of nodes representing the phases constituting the service;
- at least one oriented arc leading from one node to another representing the linking in a two-phase sequence.

9. Process according to claim 8, characterised in that the formalisation tree additionally comprises:

- at least one node followed by several arcs oriented in parallel,
- at least one node followed by several arcs oriented as a function of the choice of the following phase depending either on a condition external to the system, or on an internal condition related to the current status of the system.

10. Process according to one of claims 1 to 9, characterised in that the intermediate model developed comprises the formalised execution flows characterising the behaviour of software entities and their interactions, at least one resource specification

specifying the physical hardware, and at least one environment specification representing user behaviour.